AMENDMENTS TO THE SPECIFICATION

Replace the paragraph beginning on page 8, line 34, with the following:

Electronic communication device 100 includes an interface conversion block 116, a modulator/demodulator (modem) processor ~~102a~~102a, an optional configurable modem processor 102b, memory 106 and 118, a processor 112, a channel coder/decoder (codec) processor 104, a base transceiver station (BTS) card controller ~~110a~~110a, and an ATM Utopia/HDLC 108. Processor 112 can either be a digital signal processor (DSP) or general-purpose microprocessor (GP uP). External memory 106 used for interleaving meets the following requirements for the present embodiment: 1)8 Mb SRAM; 2)18 MHz Performance; 3) Minimum 512K x 16 configuration; 4) Byte write-enables. However the present invention is well suited to alternative memory configurations, tailored for a given application.

Replace the paragraph beginning on page 13, line 20, with the following:

Communication device 101 includes a BTS card controller 110b that can be a state machine or an optional microprocessor. Bus 127 couples memory 118 and BTS card controller 10b to channel codec processor 104 and configurable modem processor ~~102a~~102a. This provides a more direct data route between memory 118 and configurable modem processor 102a and channel codec processor 104. Additionally, memory 106 is located adjacent to channel codec processor 104, and coupled to BTS card controller 110b, and provides improved communication and processing to channel codec processor 104. It also provides improved communication and processing between channel codec processor 104 and modem processor 102a. While only one configurable modem processor 102a is shown in FIG. 1C, communication device 101 is well suited to using more than one configurable modem processor, as appropriate in a given application. Communication device 101 does not have a separate conventional DSP chip like communication device 100. Rather, communication device 101 utilizes either an external general purpose microprocessor 103 or utilizes computing elements within configurable modem processor 102a and channel codec processor 104 to

perform functions traditionally provided by a conventional DSP chip. The alternative configurations and components discussed for communication device 100 are likewise applicable to communication device 101.

Replace the paragraph beginning on page 18, line 16, with the following:

By providing a high level API, a user can design his software in a top-down fashion. This enables top-level system problems to be rapidly identified and corrected before the low-level code is written. Additionally, this approach saves a significant amount of development time as is removes the need to rework low-level software to match high level changes. The programmer's model and API of FIG. 1D also provides efficient use of hardware a parallelism. Thus, the present invention provides a method and architecture that overcomes the challenging task of scheduling the many hardware resources in the complete system. This requires an efficient mechanism for communication between the hardware resources, both within a configurable processor, e.g., within configurable modem processor 102a of FIG. 1B, and between the configurable processors (e.g., between configurable modem processor 102a/codec processor 104 and the controlling processors, e.g., processor 112, BTS card controller 110a or 110b, and BTS cell controller 114 as shown in FIGS. 1B and 1C). The hardware utilization, scheduling, and maintenance are under the control of the API. By embedding these mechanisms in the API, a process can be designed in isolation, with the synchronization issues handled at only one level within the software hierarchy. This produces a system that is considerably quicker to build and more efficient in the use of hardware than one that uses many synchronization techniques within the design. Additional description on the process for providing C-based API programming guide 171 and API functions mapped to instruction set 172 is provided in co-pending patent application entitled "A METHOD FOR DESIGNING A CONFIGURATION FOR A CONFIGURABLE SPREAD SPECTRUM COMMUNICATION DEVICE," ~~Attorney Docket No. 9824-083-999,~~ U.S. patent application Ser. No. 09/772,582, filed on Jan. 29, 2001, now U.S. Patent No. 6,701,431. This related ~~application~~ patent is commonly assigned, and is hereby incorporated by reference.

Replace the paragraph beginning on page 30, line 8, with the following:

1) "A CONFIGURABLE CODE GENERATOR SYSTEM FOR SPREAD SPECTRUM APPLICATIONS," U.S. patent application Ser. No. 09/751,782, filed on Dec. 29, 2000, now U.S. Patent No. 6,567,017;

Replace the paragraph beginning on page 40, line 16, with the following:

Referring now to FIG. 4, a block diagram of encode/decode (codec) functions accommodated by the electronic communication device is shown, in accordance with one embodiment of the present invention. Codec function block 400 includes an address generator block 401 coupled to an allocator function block 402, in turn coupled to dynamically assignable scratch/buffer memory 404, such as random access memory (RAM), in the present embodiment. Dynamically assignable scratch/buffer memory 404 is coupled to each of the multiple possible configurable decoder functional planes. Allocator function block is implemented by allocator hardware block 219 of FIG. 2A, which includes state machine components and memory that are chosen and coupled in a manner to manage multiple functional planes, e.g., planes 406 and ~~410~~411. The configuration of the allocator components can vary depending upon the protocol implemented in the multiple functional planes, e.g., planes 406 and ~~410~~411. Dynamically, a single scratch/buffer memory 404 is operational to provide configuration and state information as required for configuring and sharing resources in the multiple functional planes 406 and ~~410~~411.

Replace the paragraph beginning on page 40, line 30, with the following:

The present embodiment of codec function block 400 includes a single functional kernel plane for a given data flow direction, e.g., decode functional plane ~~410~~ 411 for decode data path 408, and encode function plane 406 for encode data path 409. Each functional plane utilizes configurable codec hardware kernels to accommodate their different functions, which are described hereinafter.

Replace the paragraph beginning on page 40, line 35, with the following:

As an illustration, decode functional plane ~~410~~ 411 includes an exemplary arrangement of sub functions for a decode path 408 to translate encoded received data into a data signal, per an appropriate communication protocol for the desired application. The arrangement of functions includes, in one embodiment, a bit field extraction block ~~410~~ 411 coupled to a memory block 412, which is then coupled to deinterleaver block 414, which is in turn coupled to rate matching block 416. Rate matching block 416 is coupled in parallel to Viterbi block 418 and to Turbo decoder block 420, both of which are then coupled to provide data to cyclic redundancy checker (CRC) block 422. Lastly, CRC block 422 is coupled to buffer 424. A similar, complementally coupled arrangement of sub-functions exists on encode functional kernel plane 406, but is omitted for clarity.

Replace the paragraph beginning on page 41, line 19, with the following:

Decode function kernel plane ~~410~~ 411 includes other user-configurable decoder functions, described hereinafter, for convolution decoding and turbo decoding. The basic function of convolution decoding and turbo decoding is known by those skilled in the art. The convolution decoder function has the following user-programmable (or user-configurable) parameters: code

polynomial; code (R, K), with K=5-9, data rate; blind rate-detection for voice channels; rate matching. Convolution decoder function also has user-programmable depuncturing pattern, traceback method, and soft-decision output. Turbo decoder function has user-programmable: code polynomials (K=3,4); data rate; block size (up to 6120); number of iterations; termination conditions; decoding metric (max-log-MAP, user-specified correction table); input scaling; traceback method; and depuncturing pattern.

Replace the paragraph beginning on page 43, line 12, with the following:

While the present embodiment shows only two functional planes, the present invention is well suited to using any number of functional planes, as appropriate for a given application. In the present embodiment, functional planes 406 and ~~410~~ 411 can be configured to perform codec functions for a wide range of communication applications, as described hereinabove. For example, in one embodiment, multiple functional planes (not shown) for both encoding and/or decoding can exist. In the present embodiment, additional functional planes for a given data flow, e.g., decoding path 408, include the same functions for decoding. Thus, every functional plane in codec functions is equally configurable with the full flexibility to accommodate each of the communication protocols. In another embodiment, one or more functional planes have functional capabilities that are different from each other. For example, in one embodiment, each functional plane in codec function block 400 has the ability to handle different subsets of the superset of communication protocols accommodated by the overall communication device. Thus, one functional plane can be configured to accommodate GSM communication protocols, while another functional plane is configured to accommodate IS-136. All the functional planes can be physically implemented in a single codec hardware processor block, e.g., block 104 of FIG. 1B, in the present embodiment. However, the present invention is well suited to a wide variety of physical implementations. The configurability and dynamic reconfigurability of codec functional planes is also described in subsequent flowcharts.

Replace the paragraph beginning on page 44, line 11, with the following:

The sub functions that are performed concurrently, e.g., in parallel, are referred to as a threads, e.g., thread 1 501, thread 2 502, and thread 3 503, for the modem function A 504. The number of threads can vary in different embodiments, according to the operations required by a given function or sub function, and their need to be performed concurrently, e.g., defined by a communication protocol in the present embodiment. Concurrent operations 508 are performed on each of the respective threads ~~501-403~~501-503 in a communication device, according to the sub function requirements. Thereafter, the threads are terminated with the provision of data and control is returned the microprocessor. Each thread can be implemented on an autonomous configurable kernel in the present embodiment. A processor 112 along with an optional allocator 219, as shown in FIG. 2A, can initiate or manage the separating and combining operations.

Replace the paragraph beginning on page 45, line 9, with the following:

Flowchart 6100 begins with step 6102. In step 6102 of the present embodiment, a design configuration is received at a configurable device. For example, a design configuration for a configurable communication device can be a 3GPP-specific configuration, implemented by user-specified proprietary algorithms, in one embodiment. Additional description on the process for designing a configuration for a configurable device is provided in co-pending patent application entitled "A METHOD FOR DESIGNING A CONFIGURATION FOR A CONFIGURABLE SPREAD SPECTRUM COMMUNICATION DEVICE~~.~~," Ser. No. 09/772,582, filed Jan. 29, 2001, now U.S. Patent No. 6,701,431.